NASA TM- 86412

NASA Technical Memorandum 86412 NASA-TM-86412 19850017209

CRYSTALLINITY DETERMINATION BY CURVEFIT PROCEDURE FOR A SEMI-CRYSTALLINE POLYMER

N. T. WAKELYN

MAY 1985

NASA
National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665

SUMMARY

Wide angle x-ray scattering (WAXS) data from poly(etheretherketone) (PEEK) has been resolved into a crystalline contribution represented as 4 reflections and an amorphous contribution represented as a broad, smoothly varying curve, both contributions occurring in the 2θ range: 15-31 degrees. In this resolution the crystalline scatter is described as a linear combination of Cauchy and Gaussian functions while that of the amorphous halo is expressed as a cubic polynomial. Statistical analysis of the measured scattered intensity from an amorphous specimen with that calculated from the cubic polynomial, as a function of the combination parameter (fraction of Cauchy and Gaussian functions), suggests that the crystalline fraction of the polymer specimen studied is about 0.39. A listing of the FORTRAN IV program used in the resolution is provided in the Appendix.

INTRODUCTION

Composites fabricated from thermoplastic resins have some inherent differences in physical properties from those of the current thermoset resin composites and some formulations show considerable promise for structural applications. A partial listing of some of the potential advantages of such materials should include ease of formability, damage repair, and bonding as well as good environmental resistance and re-processing capability.[1-3] The concept of a thermoplastic implies the possibility of some degree of crystallization and with this possibility there is the promise of material property enhancement. Indeed, increased toughness, reduced moisture

N85-25520#

degradation, and the chemical resistance required for use in aerospace structures have been associated with crystallization in thermoplastic resin matrices.[4]

Poly(etheretherketone) (PEEK) is one of the thermoplastic matrix resins which can be fabricated as a semi-crystalline matrix in combination with a carbonaceous fiber reinforcement. This material has been characterized as having mechanical properties superior to those of currently used epoxies when tested under wet conditions at elevated temperatures.[4] Physical property data and details of the synthesis of PEEK and other polyaryletherketones have been reported.[5,6]

In order to understand the relationship between a desired physical property and an induced crystalline order in a matrix material a method for estimation of the degree of this order is necessary. Added complexity in a polymeric structure, e.g. changing from a one to a two phase material, is sure to have a far from straightforward effect on desired physical properties. It is well within reason to expect a small amount of added crystalline order to produce an increase in toughness, while a greater increase yields a brittle material. Hopefully a method for measuring crystalline order in a polymer matrix would also be useful in the presence of a carbonaceous reinforcement.

Wide angle x-ray scattering (WAXS) has been used to investigate the nature of the crystalline structure appropriate for the poly(arylether) and poly(aryletherketone) polymers.[7-9] The basis for the interpretation of the WAXS data for PEEK and the other poly(aryletherketone) polymers has been the study of poly(p-phenylene oxide) (PPO) which is reported[7] to crystallize in the space group Pbcn (space group No. 60, International Tables for X-ray

Crystallography). While PEEK is a more complicated polymer than PPO and requires at least six aryl units aligned along the c-axis of its unit cell for its description,[6,8] the symmetry elements of Pbcn are appropriate for PEEK, as well as for PPO, and the major peaks in the WAXS data for this material can be indexed based upon the assumption of a structural analogy between the two materials.[10] The four major crystalline reflections of PEEK, occurring at ~18.7, 20.7, 22.6, and 28.7° (2θ), have, therefore, been indexed as 110, 113, 200, and 213, respectively, in strict analogy with the indexing of reference 9.

The purpose of this report is to describe an analytical procedure for the estimation of degree of crystalline order in a thermoplastic polymer matrix. Semicrystalline PEEK, a material of possible future use in aerospace applications, has been used to develop and illustrate this procedure. Intensity versus diffraction angle data taken from a flat powder specimen in a WAXS experiment has been used for the analysis. In order to build in a certain realism and thus to facilitate extention to real specimens of composites reflection geometry has been used to obtain the data. While the computer program developed for the analysis and presented in the Appendix is specialized for PEEK it is readily convertible to other formulations for which the major x-ray diffraction reflections are known.

ANALYTICAL

Various methods for the determination of degree of crystalline order in polymeric materials using WAXS data have been employed, some of which have achieved an almost standard practice status. A sampling of these methods,

in order of increasing ease of application, vary from a difficult to use but sophisticated absolute method,[11] through a more practical though less rigorous absolute procedure,[12] through a well-regarded index procedure,[13,14] to a surprisingly useful 2-point index method.[15] All of these would be confounded by the presence of the carbonaceous reinforcement and involve factors such as a requirement for multiple samples, transmission geometry, interpolation between standards, and for some, even an element of subjectivity. For those polymers with known crystal structure, however, a crystallinity determination by curvefit procedure[16] is feasible and in theory, at least, is unaffected by the previously mentioned factors and may even be practicable in the presence of the carbonaceous reinforcement.

The basis of the curvefit procedure[16] is the expression of the measured x-ray intensity data versus diffractometer angle (2θ) for the polymeric specimen as a sum of the x-ray scatter from the known crystalline peaks plus that characteristic of the broad, smoothly varying paracrystalline or amorphous scatter. The crystalline peaks are described as a linear combination of Cauchy and Gaussian expressions with the linear combination parameter varying from 0 for a pure Cauchy function to 1 for a pure Gaussian function with intermediate values representing a combination of the 2 functions. The amorphous background scatter is expressed as a cubic polynomial.

Each crystalline peak is defined by parameters representing the peak height, peak position, and width of the peak at half maximum, while the cubic polynomial is defined by 4 parameters. Thus, the expression for the summation of the crystalline peaks plus that of the background curve for the 4 reflections representative of PEEK in the 2θ range of interest contains 16

independent variables to be determined by the curvefit procedure. These independent variables were determined by solving the system of nonlinear equations, i.e. the summation of the expressions for the crystalline reflections and background curve as a function of the diffraction angle minus the measured intensity value at that angle, by using Levenberg-Marquardt and Gauss algorithms for the nonlinear least squares approximations.[17] The computer program, which solves for these variables and which contains the Langley computing center's library routine, MARQ,[18] is presented in the Appendix for a real case. Convergence of the modified Marquardt algorithm is satisfied when the difference of the residual sum of squares estimates on two successive iterations divided by the residual at the first of these iterations is equal to as less than an input parameter based upon the relative accuracy of the equations.[18]

Once the parameters have been obtained, an integration or comparable mathematical procedure may be used to extract the relative areas under the various curves. An obvious first approximation to the polymeric crystalline fraction would then be the summation of the areas under the crystalline reflections divided by the area under all curves. There is apparently some controversy about this, however, since some workers[12] report different constants of proportionality for relating the crystalline and amorphous fractions to the respective integral intensities (areas). Other workers,[19] base their data reduction upon the "Law of Conservation of Intensity",[20] in effect, choosing the more simple procedure. Since the present goal is the description of an analytical procedure as applied to a specific thermoplastic material and since only one crystalline specimen was required

to develop and illustrate this procedure, the more simple expression for crystalline fraction is presented.

EXPERIMENTAL

WAXS data has been obtained on annealed and melt-quenched PEEK material. CuKα radiation was used with an automated powder x-ray diffractometer equipped with a curved crystal, graphite monochromator. With the generator operated at 45 kV and 40 mA, the intensity of 1 s counts taken every 0.01 degrees (2θ) was recorded on hard disk. The data in the angular range 15.00-30.83°, an adequate range for evaluation of crystalline fraction,[12] was used for the present analysis.

Commercially synthesized polymeric material, received in the form of compression mouldings, was melted in a hot press and immediately quenched in ice water to produce a light brown, transparent film which had a density of 1.267 g cm^{-3} as determined by immersion in a toluene-carbon tetrachloride density gradient column. This film was cooled to dry ice temperature, ground in a rotary mill, and sieved using a 60 mesh screen. Part of the material which passed through the screen was used as is (the melt-quenched specimen) and part was annealed in a vacuum oven at 267°C for 72 h to induce crystallinity. The density of this annealed material (the crystalline specimen) was determined to be 1.295 g cm^{-3}.

RESULTS AND DISCUSSION

The mathematical expression used to describe the crystalline reflections contains a linear combination parameter which controls their

geometrical shape. When this combination parameter has the value 0, the reflections are characterized as pure Cauchy functions, and when its value is 1 they are represented as pure Gaussians. Intermediate values of the parameter produce composite representations of the reflections. Figure 1 presents a series of 10 plots (Figures 1(a) - 1(k)) depicting the changes incurred in the crystalline peaks and the background curve as the linear combination parameter is varied from 1.0 to 0.0 in 0.1 increments. They are compared on each plot with the actual measured data of semi-crystalline PEEK appearing at the top of the figure as a curve composed of a series of points over the angular range 15.00-30.83° (2θ). The indices describing the crystalline reflections are included in Figure 1(a).

As the description of the crystalline peaks changes from pure Gaussian (Fig. 1(a)) to pure Cauchy (Fig. 1(k)) tailing at the base of the peaks increases uniformly. This increased tailing produces a directly related increase in the area of the crystalline peaks, which increase at the expense of the background area. The fractional crystalline area, which is the sum of areas under the 4 crystalline peaks divided by this sum plus the area under the background curve, varies with the increased tailing from 0.2502 to 0.3916 in arbitrary intensity-degree units. This fractional crystalline area is related to a degree of crystalline order with the assumption of equivalent scatter from equivalent phases. In all cases there is insignificant difference between the calculated area under the measured curve and the calculated summation of all curvefit areas.

Since the summation of crystalline areas plus the area under the background curve is essentially constant, and since the summation of the crystalline areas changes with the linear combination parameter, the shape

of the background curve (as well as its area) also changes with this parameter. A comparison of this background curve with that measured for amorphous PEEK, normalized for equal area, is presented in Figure 2, as a function of the linear combination parameter. There is a progressive observable difference in the relationship between the 2 compared curves presented in Figures 2(a) and 2(b); however, a statistical analysis was performed to minimize the necessity for subjective judgment as to the quality of a representation. The simple correlation coefficients between the set of points describing the background curve as a function of the linear combination parameter and the measured data for an amorphous specimen, each having 1584 observations, were determined and are presented in Table I.

The variation of the amorphous area, fractional crystalline area, and correlation coefficient as the form of the crystalline peak shape is systematically varied from that of a pure Cauchy function to that of a pure Gaussian function is presented in Table I. There is displayed a systematic increasing trend in the correlation coefficient with increasing Cauchy character suggesting that for the present WAXS experiment with semi-crystalline PEEK powder the pure Cauchy description of the crystalline peak shape is preferred. Others, however, have reported a linear combination parameter of 0.5 to yield the best fit for many synthetic and natural fibers,[16] while another worker analyzed nylon data using pure Cauchy expressions.[21] Concomitant with the present trend is the determination that the crystalline fraction of the annealed PEEK is 0.39.

CONCLUDING REMARKS

A curvefit procedure, based upon the modified Marquardt algorithm, has been developed and illustrated with semi-crystalline PEEK. Intensity versus diffraction angle data taken from flat powder specimens (crystalline and amorphous) in reflection geometry has been used for the analysis. While the method was designed to be based upon one semi-crystalline sample only, an amorphous specimen was required to discriminate between the various solutions. Extention of the analytical procedure (and computer program) to other systems, such as PEEK composites containing carbonaceous fiber, should be straightforward.

REFERENCES

1. J. T. Hoggatt, 5th National SAMPE Technical Conference, 5, 91 (1973).
2. M. G. Maximovich, 19th National SAMPE Symposium and Exhibition, 19, 262 (1974).
3. J. T. Hoggatt, 20th National SAMPE Symposium and Exhibition, 20, 606 (1975).
4. J. T. Hartness, 14th National SAMPE Technical Conference, 14, 26 (1982).
5. T. E. Attwood, P. C. Dawson, J. L. Freeman, L. R. J. Hoy, J. B. Rose and P. A. Staniland, Polym. Prepr. 20, 191 (1979).
6. T. E. Attwood, P. C. Dawson, J. L. Freeman, L. R. J. Hoy, J. B. Rose and P. A. Staniland, Polymer, 22, 1096 (1981).
7. J. Boon, and E. P. Magre, Makromol. Chem., 126, 130 (1969).
8. P. C. Dawson and D. J. Blundell, Polymer Reports, 21, 577 (1980).

9. D. R. Rueda, F. Aria, A. Richardson, I. M. Ward and F. J. Balta Calleja, Polymer Communications, 24, 258 (1983).

10. N. T. Wakelyn, Polymer Communications, 25, 306 (1984).

11. W. Ruland, Acta Cryst., 14, 1180 (1961).

12. G. Challa, P. H. Hermans, and A. Weidinger, Makromol. Chem., 56, 169 (1962).

13. J. H. Wakelin, H. S. Virgin, and E. Crystal, J. Appl. Phys., 30, 1654 (1959).

14. W. O. Statton, J. Appl. Polymer Sci., 7, 803 (1963).

15. D. E. Bosley, J. Appl. Polymer Sci., 8, 1521 (1964).

16. A. M. Hindeleh and D. J. Johnson, J. Phys. D. Appl. Phys., 4, 259 (1971).

17. K. M. Brown and J. E. Dennis, Numerische Mathematik, 18, 289 (1972).

18. Mathematical and Statistical Software at Langley, Central Scientific Computing Complex Document N-3, April 1984.

19. A. M. Hindeleh and D. J. Johnson, Polymer, 11(12), 666 (1970).

20. B. K. Vainshtein, Diffraction of X-Rays by Chain Molecules, Page 178, Elsevier Publishing Co., London, 1966.

21. J. O. Warwicker, J. Soc. Dyers Colour., 86, 303 (1970).

Table I. Results of Curvefitting Procedure and Statistical Analysis as a Function of the Linear Combination Parameter for Semi-crystalline PEEK Polymer Matrix.

COMBINATION PARAMETER*	AMORPHOUS AREA**	FRACTIONAL CRYSTALLINE AREA	CORRELATION COEFFICIENT***
0.0	23441	.3916	.9705
0.1	24079	.3751	.9701
0.2	24690	.3592	.9698
0.3	25276	.3440	.9694
0.4	25842	.3293	.9692
0.5	26388	.3151	.9690
0.6	26917	.3014	.9687
0.7	27431	.2881	.9685
0.8	27930	.2751	.9683
0.9	28416	.2625	.9681
1.0	28891	.2502	.9679

*A linear combination parameter of 0.0 implies a pure Cauchy function while that of 1.0 implies a pure Gaussian expression.
**Area under amorphous halo in intensity - degree units relative to an area under the measured intensity envelope of 38530.
***A measure of how well the shape of the derived cubic polynomial, describing the amorphous halo, is correlated to the measured shape of an amorphous PEEK specimen.

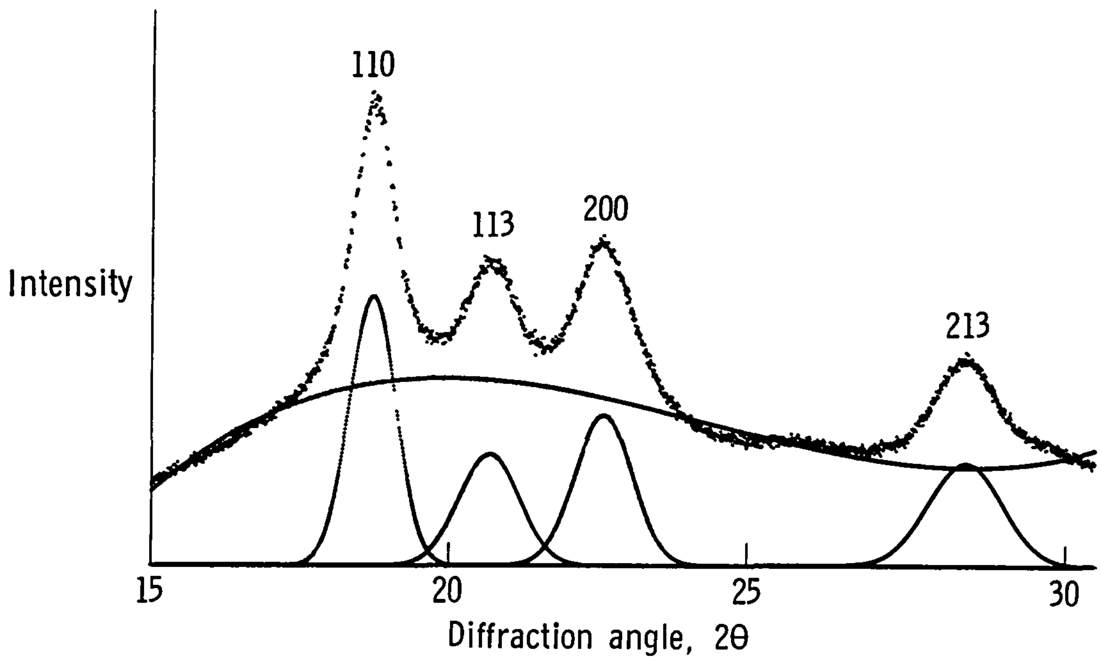

Figure 1(a). - X-ray diffractogram of semi-crystalline PEEK compared to resolved crystalline scatter represented as a linear combination of Cauchy and Gaussian functions and the amorphous halo represented by a cubic polynomial. Combination parameter = 1.0. Fractional crystalline area = 0.2502.

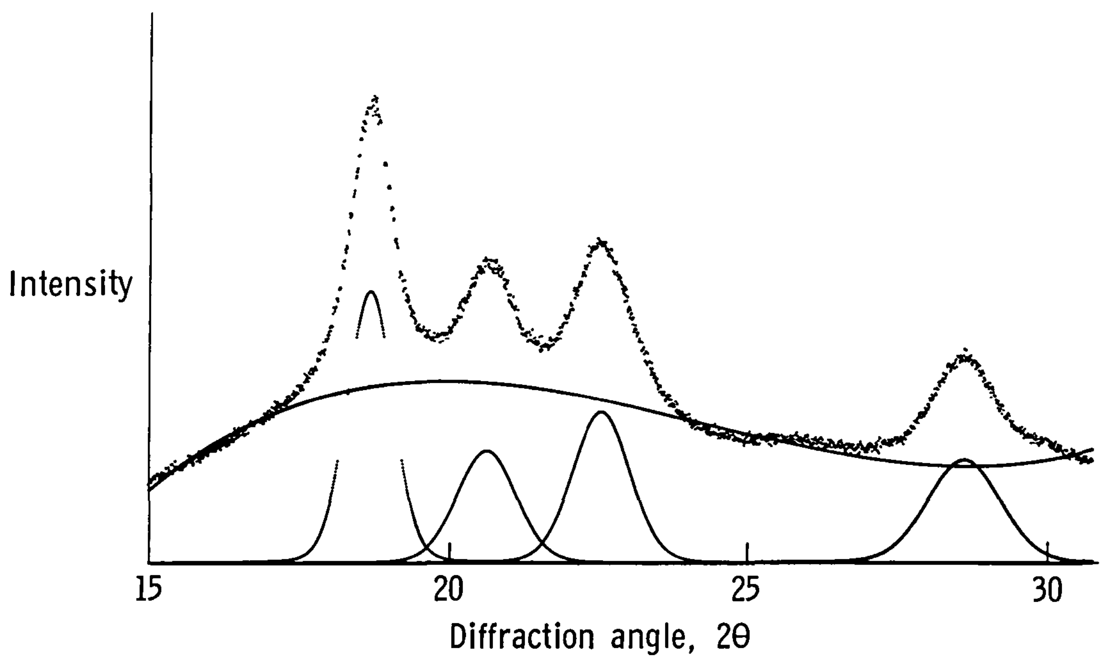

Figure 1(b). - Combination parameter = 0.9. Fractional crystalline area = 0.2625.

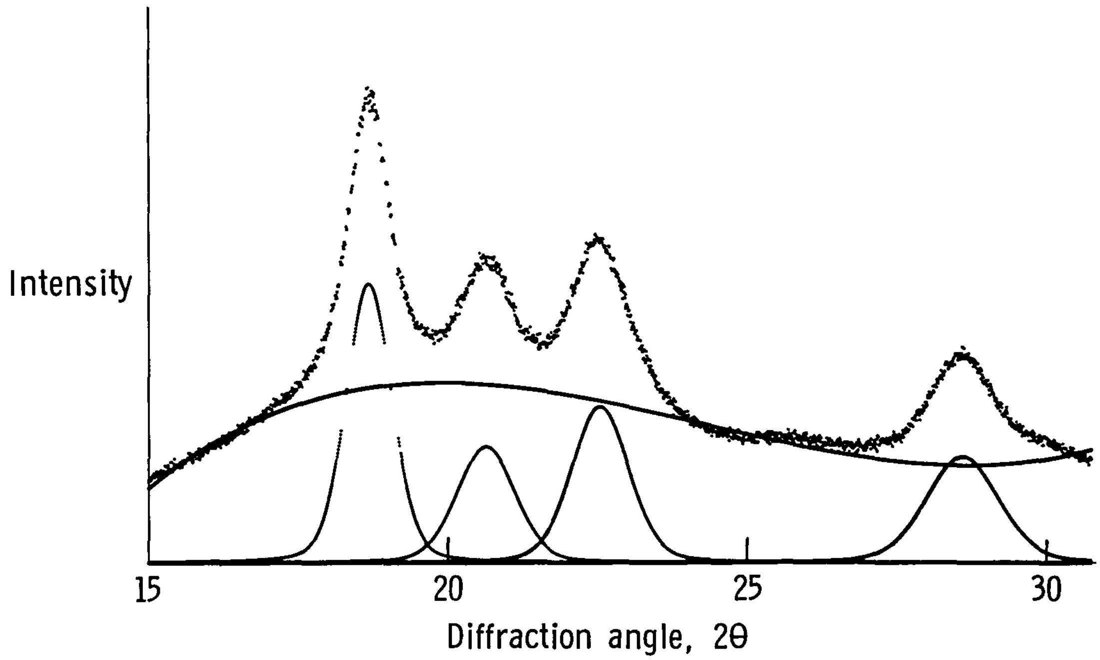

Figure 1(c). - Combination parameter = 0.8. Fractional crystalline area = 0.2751.

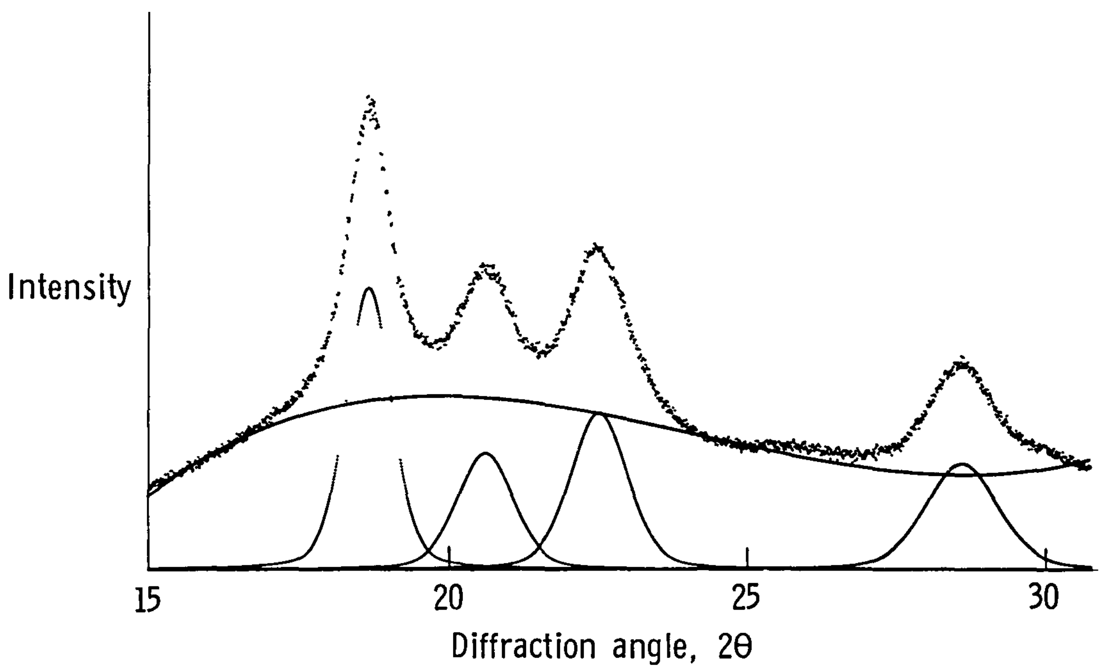

Figure 1(d). - Combination parameter = 0.7. Fractional crystalline area = 0.2881.

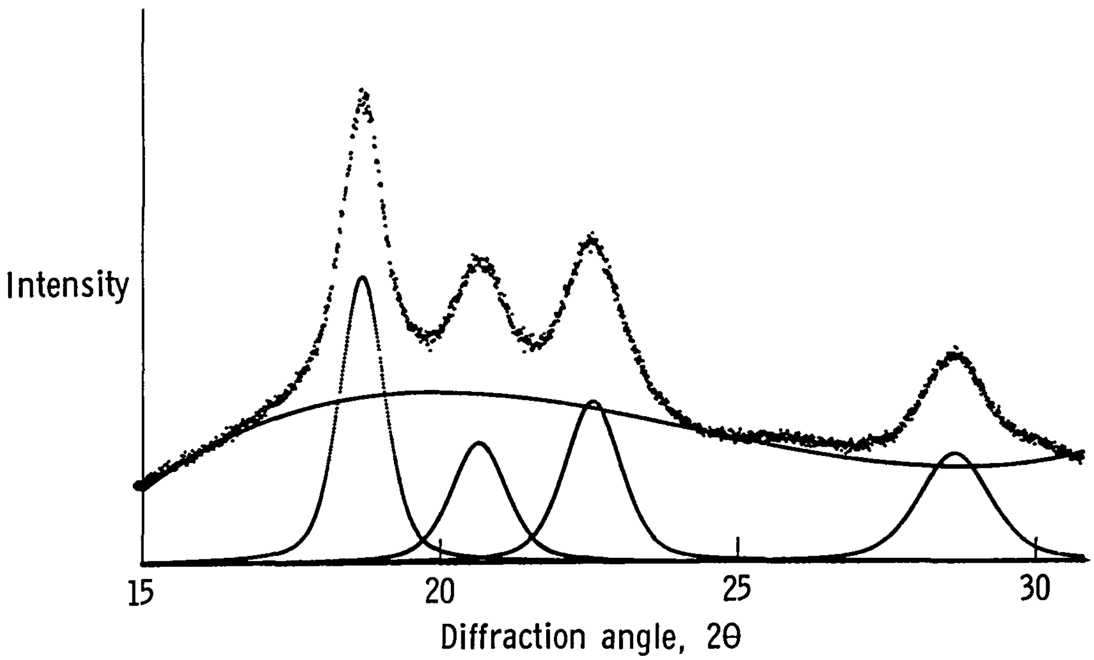

Figure 1(e). - Combination parameter = 0.6. Fractional crystalline area = 0.3014.

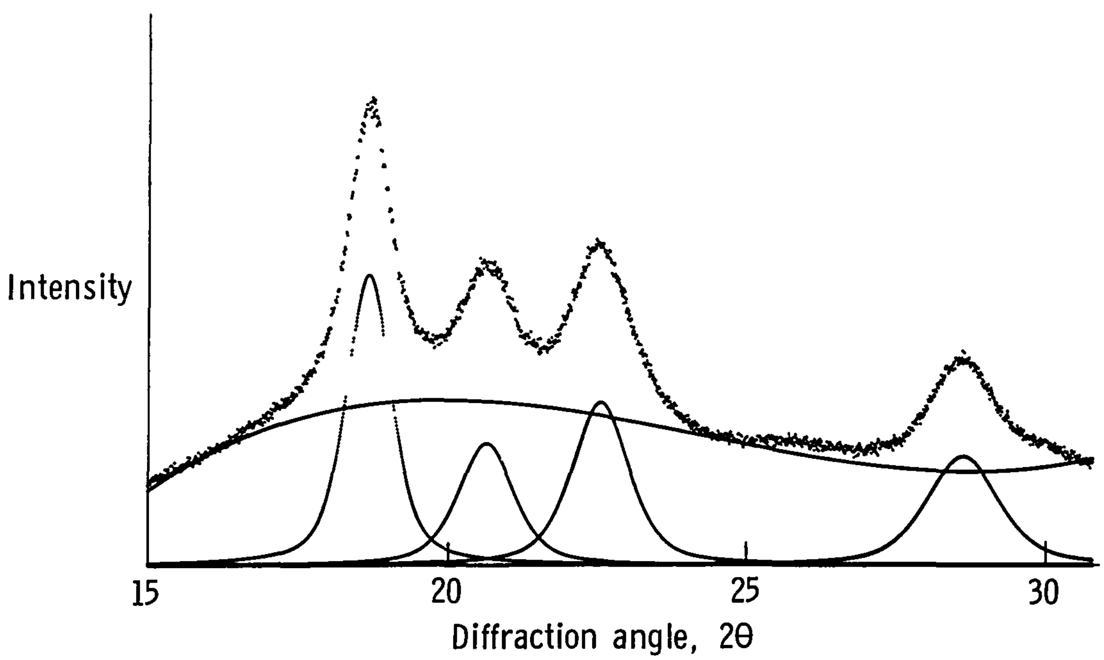

Figure 1(f). - Combination parameter = 0.5. Fractional crystalline area = 0.3151.

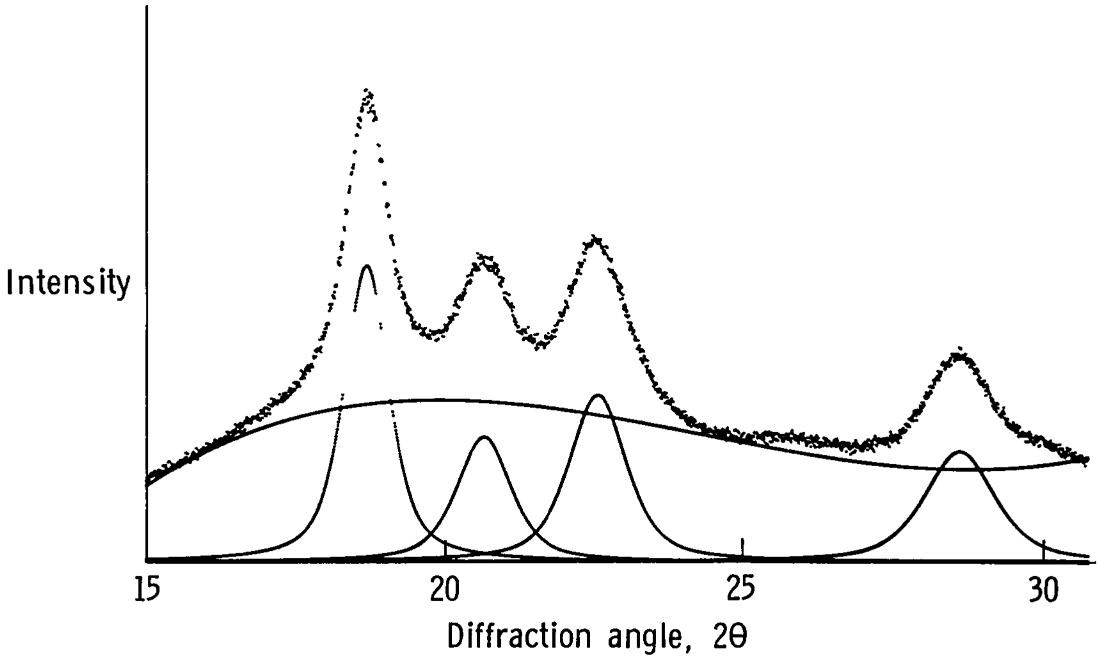

Figure 1(g). - Combination parameter = 0.4. Fractional crystalline area = 0.3283.

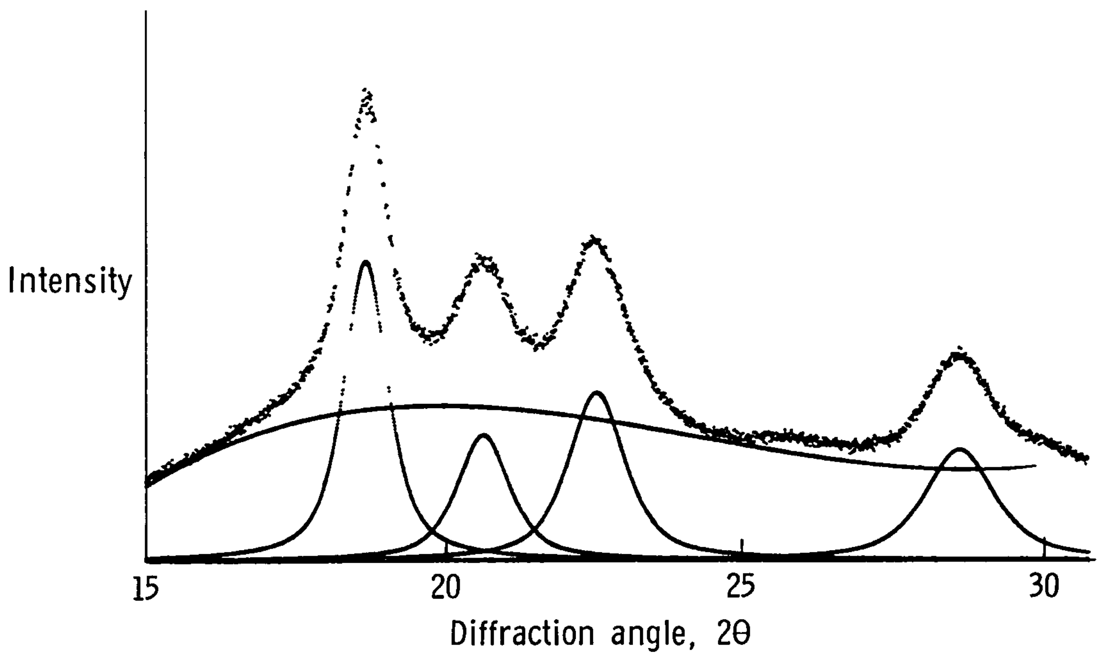

Figure 1(h). - Combination parameter = 0.3. Fractional crystalline area = 0.3440.

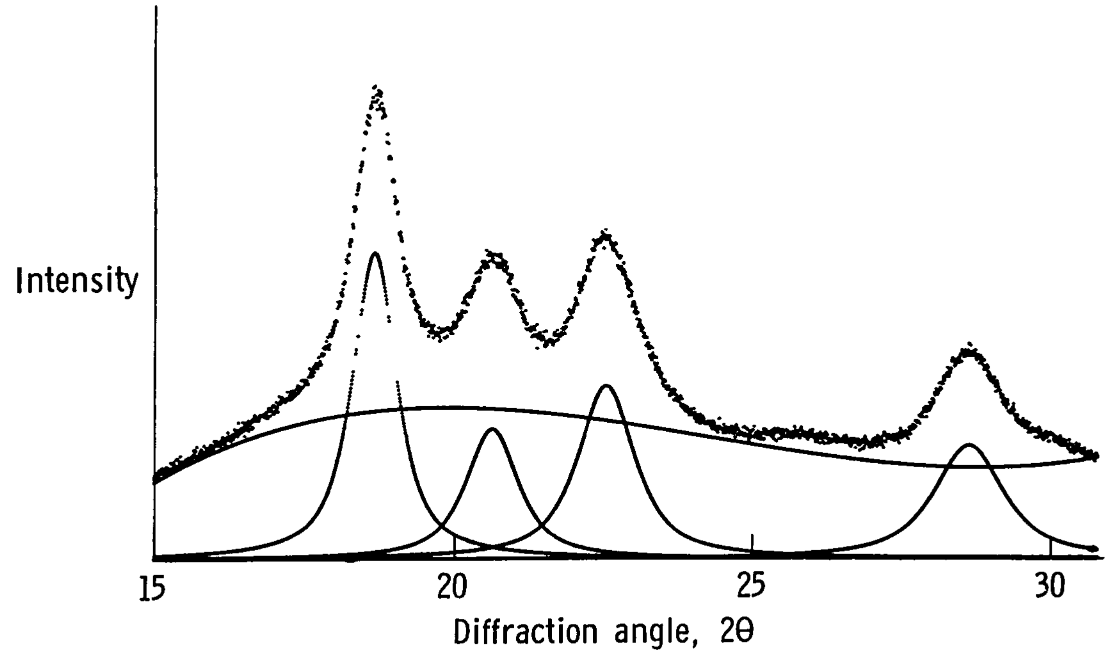

Figure 1(i). - Combination parameter = 0.2. Fractional crystalline area = 0.3592.

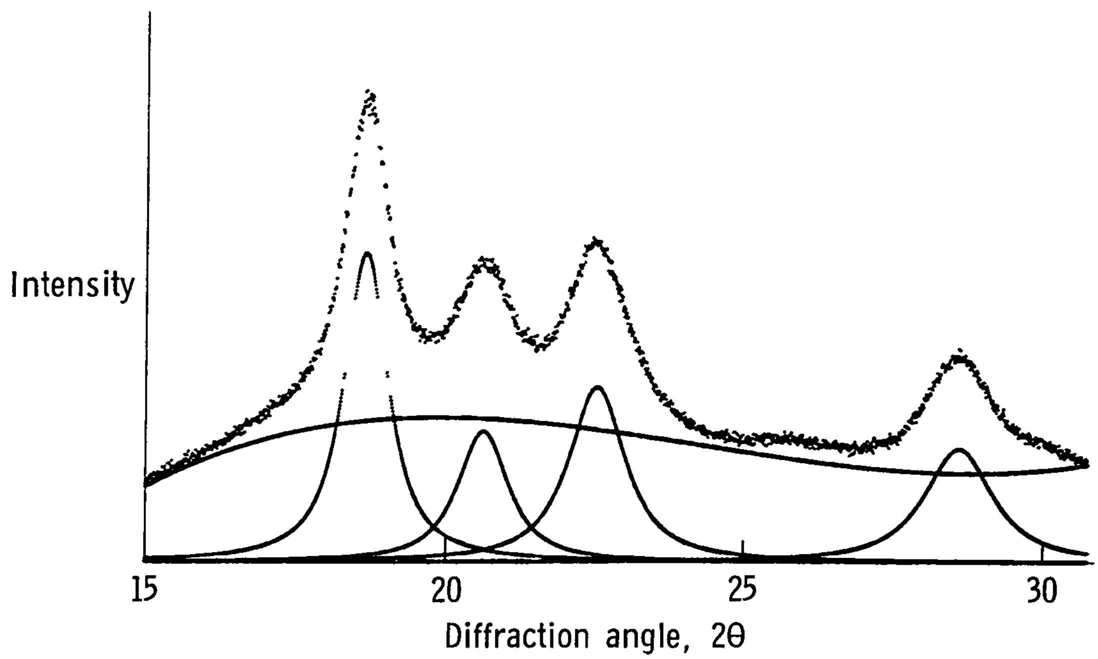

Figure 1(j). - Combination parameter = 0.1. Fractional crystalline area = 0.3751.

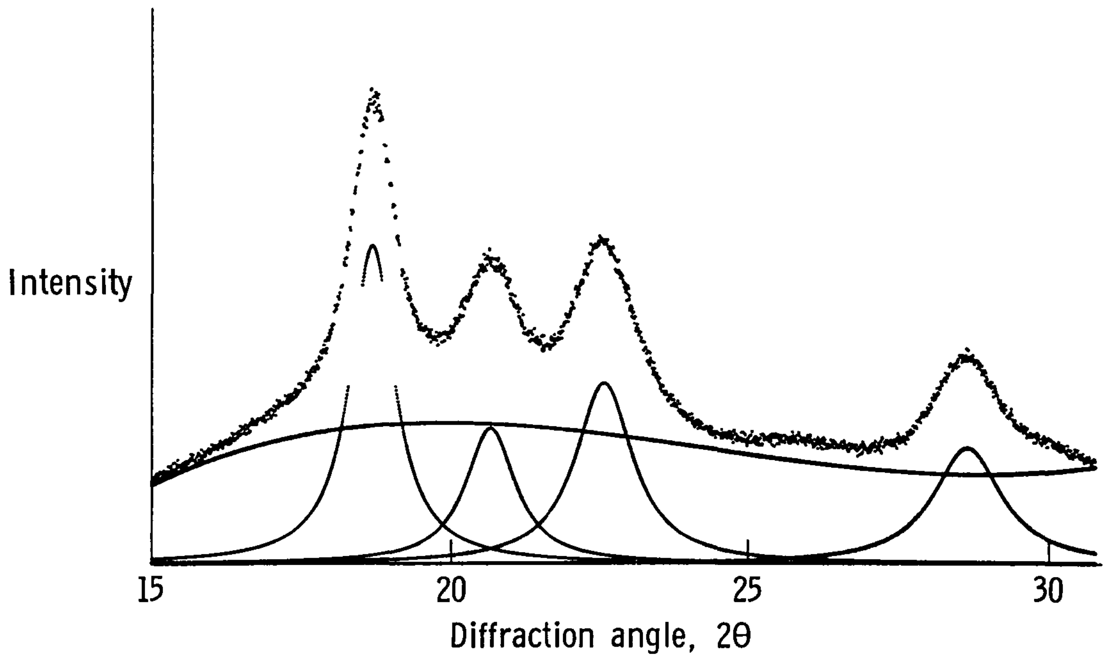

Figure 1(k). - Combination parameter = 0.0. Fractional crystalline area = 0.3916.

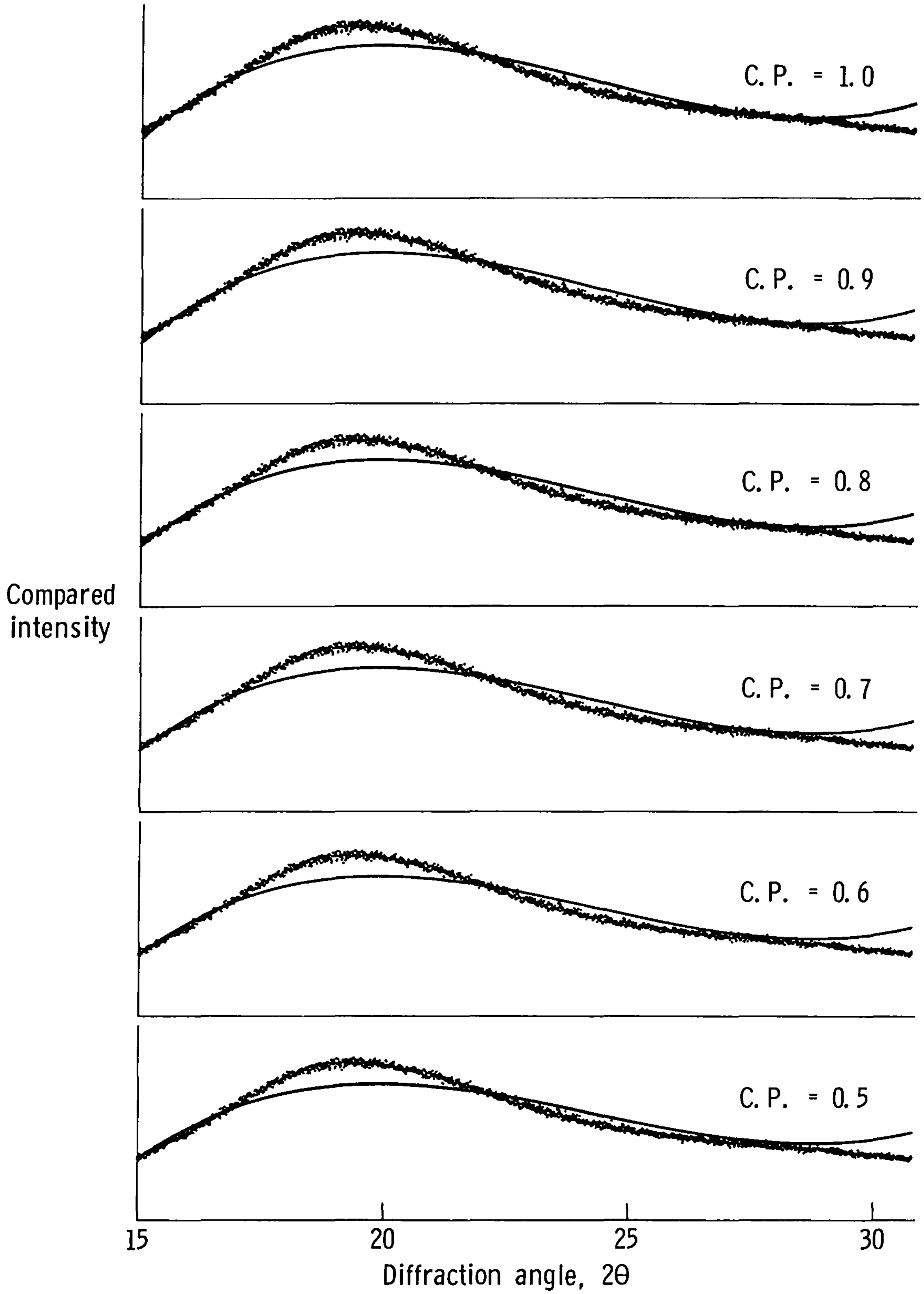

Figure 2(a). - Amorphous halo as represented by a cubic polynomial compared with data from amorphous PEEK, normalized for equal area, for combination parameters varying from 0.5 to 1.0.

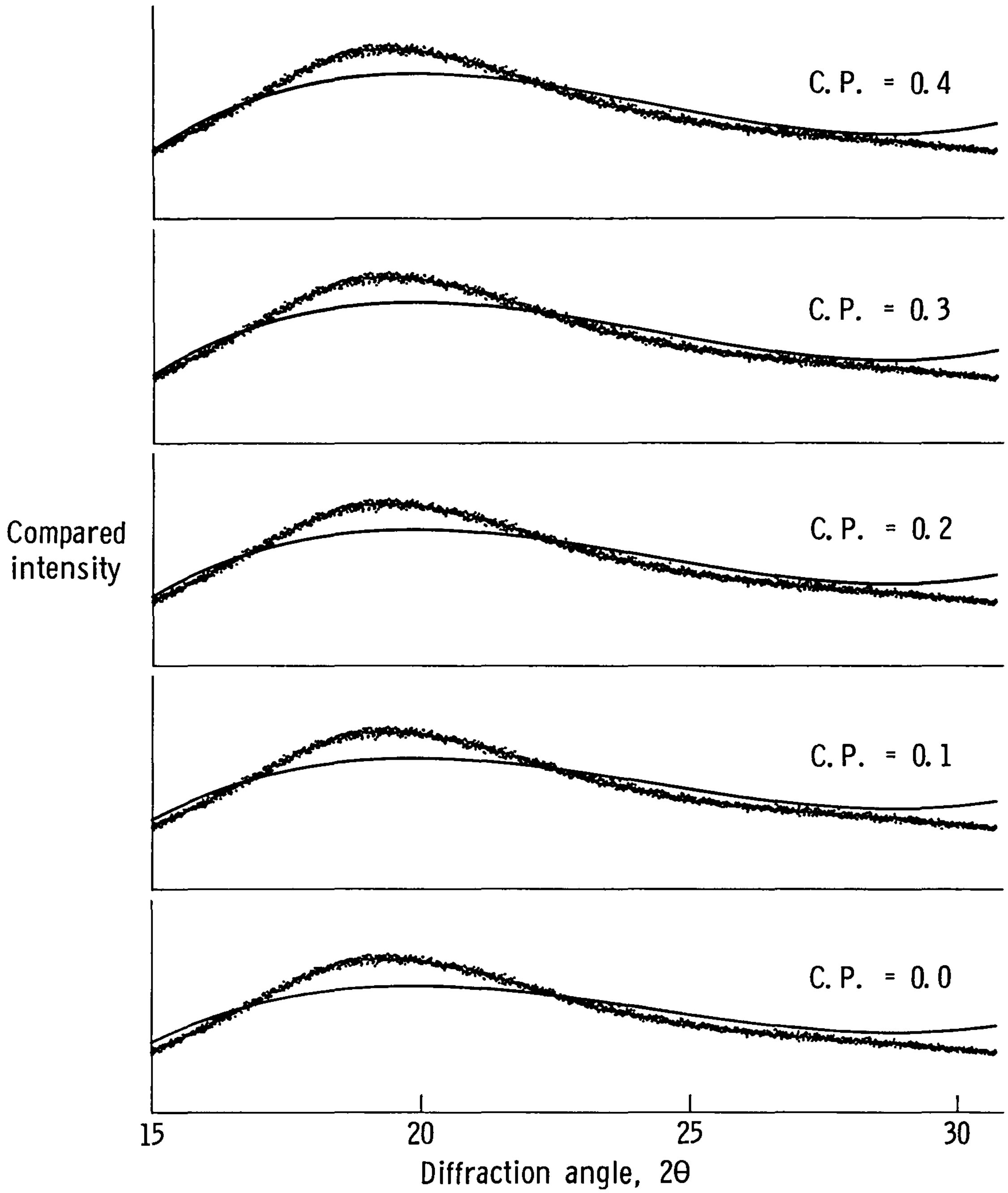

Figure 2(b). - Amorphous halo as represented by a cubic polynomial compared with data from amorphous PEEK, normalized for equal area, for combination parameters varying from 0.0 to 0.4.

APPENDIX I. A FORTRAN IV COMPUTER PROGRAM FOR THE RESOLUTION OF POLYMER MATRIX CRYSTALLINE SCATTER

The code listed in this Appendix represents that used to calculate one of the cases presented in this report. A dummy call to a PLOT routine is used since these routines are, in general, arbitrary and thus not transportable. Subroutine MARQ, which solves a system of N nonlinear equations in N unknowns for the real roots by the modified Marquardt algorithm, is a Langley Research Center library routine. (Mathematical and Statistical Software at Langley, Central Scientific Computing Complex Document N-3, April 1984).

```
      PROGRAM MARFIT(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
      EXTERNAL F
      COMMON TT, XRI, G
       DIMENSION X(16), FX(1584), WK(33800), TT(1584), XRI(1584)
      DIMENSION IRAY(6)
        DIMENSION IN(24)
      DATA IRAY/6 * (-0)/
      DATA M/1584/, N/16/, C/1.0/, EPS/1.0E-08/, NSIG/5/, ITMAX/100/
      DATA X/ 2766.,              18.71,              1.0,
     1          952.,             20.67,              1.0,
     2         1570.,             22.56,              1.0,
     3         1008.,             28.65,              1.0,
     5        -9500.,             1300.,              -50.,
     6         0.600/
COMMENT.   INITIALIZE G AT THIS TIME
      G = 0.5
COMMENT: OBTAIN X-RAY INTENSITIES FROM ARBITRARY FILE
      DO 5 I=1,M
      READ(5,3) IN
    3 FORMAT(8(A1,A5,A4))
C     IF(IN(2) .EQ. 5H10.00) GO TO 6
      IF(IN(2) .EQ. 5H14.98) GO TO 6
    5 CONTINUE
    6 PRINT4,IN(2)
    4 FORMAT(1X,A5)
      I1 = 1
      I2 = I1 + 7
      DO 25 I3 = 1,6
      DO 50 I4 = 1,31
      READ(5,7) (XRI(I),I=I1,I2)
    7 FORMAT(10X,8(F7.0))
      I1 = I1 +8
      I2 = I2 + 8
   50 CONTINUE
      I2 = I2 - 6
      READ(5,8) (XRI(I),I=I1,I2)
    8 FORMAT(10X,2(F7.0))
      I1 = I1 + 2
      I2 = I1 + 7
   25 CONTINUE
      DO 26 J=1501,1584,8
      KEND=J+7
      IF(J .EQ. 1581)KEND=1584
      READ(5,7) (XRI(I),I=J,KEND)
   26 CONTINUE
         PRINT*,KEND
COMMENT: WE NOW HAVE 1584 XRI'S
      IRAY(4) = 0
      CALL SYSTEMC (115,IRAY)
      PRINT1
    1 FORMAT(1H1,18X,17HEXPERIMENTAL DATA)
      PRINT2
    2 FORMAT(1H0,13X,9H 2-THETA ,10X,9HINTENSITY,//)
COMMENT: GENERATE 2-THETA'S CORRESPONDING TO XRI'S
```

```
      T = 14.99
      DO 100 I=1,M
      T = T + 0.01
      TT(I) = T
  100 CONTINUE
  105 CONTINUE
COMMENT: MARQ USES GAUSS AND LEVENBERG-MARQUARDT ALGORITHMS
C   TO RESOLVE WAXS INTO CRYSTALLINE AND PARACRYSTALLINE CONTRIBUTIONS
      CALL MARQ(M,N,X,C,EPS,NSIG,ITMAX,FX,F,WK,IERR)
      PRINT150,G
  150 FORMAT(1H0,3HG =,G22.14)
      PRINT900,IERR
      IF(IERR .NE. 0) GO TO 10
C     WRITE(6,901) (X(I),I=1,N), (FX(I),I=1,M), ITMAX
      PRINT200
  200 FORMAT(1H1,25X,15HSOLUTION VECTOR,//)
      PRINT250
  250 FORMAT(1H0,5X,11HPEAK HEIGHT,10X,13HPEAK POSITION,7X,17HWIDTH AT 1
     1/2 MAX.,//)
      NX = N - 4
      PRINT300, (X(I),I=1,NX)
  300 FORMAT (1H ,3G22.14)
      PRINT325
  325 FORMAT(1H0,35X,23HPOLYNOMIAL COEFFICIENTS)
      NP = NX+1
      PRINT350, (X(I), I=NP,N)
  350 FORMAT(1H0,4G22.14)
      PRINT400
  400 FORMAT(1H1,30X,19HVALUES OF EQUATIONS,////)
      PRINT300, (FX(I),I=1,M)
      PRINT500,ITMAX
  500 FORMAT(23H0NUMBER OF ITERATIONS =,I4)
      CALL AREA(X,M,N)
      CALL PLOT(X,M,N)
      GO TO 20
   10 WRITE(6,902) (X(I),I=1,N)
      PRINT500,ITMAX
   20 STOP
  900 FORMAT(13H0ERROR CODE =,I4)
  901 FORMAT(18H0SOLUTION VECTOR =,3G22.14/22H0VALUES OF EQUATIONS =,
    1        4G22.14/23H0NUMBER OF ITERATIONS =,I3)
  902 FORMAT(33H0LAST APPROXIMATION TO SOLUTION =,3G22.14)
      END
      SUBROUTINE PLOT(X,M,N)
      RETURN
      END
      SUBROUTINE F(X,M,N,FX)
      COMMON TT, XRI, G
      DIMENSION X(N), FX(M), TT(1584), XRI(1584)
      DIMENSION PH(4), PP(4), PW(4), EQ(4)
      A1 = X(N-3)
      A2 = X(N-2)
      A3 = X(N-1)
      A4 = X(N)
```

```
      DO 100 I=1,M
      INDEX = 0
      DO 200 L=1,4
      PH(L) = X(L+INDEX)
      PP(L) = X(L+1+INDEX)
      PW(L)  = X(L+2+INDEX)
      INDEX = INDEX + 2
          EQ(L) =  (G *PH(L) * EXP(-ALOG(2.0)*((2.0*(TT(I)- PP(L)) / PW(
     1  L))**2.0)))+ (((1.0-G)*PH(L))/ (1.0+(((2.0*(TT(I)- PP(L))) / PW(
     2  L))**2.0)))
  200 CONTINUE
      B = A1 + (A2*TT(I)) + (A3*(TT(I)**2)) + (A4*(TT(I)**3))
   50 CONTINUE
      FX(I) = EQ(1) + EQ(2) + EQ(3) + EQ(4) - XRI(I) + B
  100 CONTINUE
      RETURN
      END
      SUBROUTINE AREA(X,M,N)
      COMMON TT, XRI, G
      DIMENSION TT(1584), XRI(1584), CP(1584)
      DIMENSION PH(4), PP(4), PW(4), EQ(1584,4)
      DIMENSION IRAY(6)
      DIMENSION B(4), X(16)
      DATA IRAY/6 * (-0)/
      IRAY(4) = 0
      CALL SYSTEMC (115,IRAY)
      DO 110 I=1,M,8
      PRINT6,XRI(I),XRI(I+1), XRI(I+2),XRI(I+3),XRI(I+4),XRI(I+5),XRI(I+
     26),XRI(I+7)
    6 FORMAT(1H ,10X,8F10.2)
  110 CONTINUE
      DO 111 I=1,M,8
      PRINT6,TT(I),TT(I+1),TT(I+2),TT(I+3),TT(I+4),TT(I+5),TT(I+6),TT(I+
     27)
  111 CONTINUE

      B1 = X(N-3)
      B2 = X(N-2)
      B3 = X(N-1)
      B4 = X(N)
      DO 150 I=1,M
      INDEX = 0
      DO 200 L=1,4
      PH(L) = X(L+INDEX)
      PP(L) = X(L+1+INDEX)
      PW(L)  = X(L+2+INDEX)
      INDEX = INDEX + 2
        EQ(I,L) =  (G *PH(L) * EXP(-ALOG(2.0)*((2.0*(TT(I)- PP(L)) / PW(
     1  L))**2.0)))+ (((1.0-G)*PH(L))/ (1.0+(((2.0*(TT(I)- PP(L))) / PW(
     2  L))**2.0)))
  200 CONTINUE
      CP(I) = B1   + B2   *TT(I) + B3   *(TT(I)**2) + B4   *(TT(I)**3)
  150 CONTINUE
```

```
COMMENT   XRI=X-RAY INTENSITY;   TAE=TOTAL AREA UNDER ENVELOPE
      TAE = 0.0
      A1 = 0.0
      A2 = 0.0
      A3 = 0.0
      A4 = 0.0
      A5 = 0.0
      PC = 0.0
      DO 225 I=1,M
      TAE = TAE + XRI(I)
      A1 = A1 + EQ(I,1)
      A2 = A2 + EQ(I,2)
      A3 = A3 + EQ(I,3)
      A4 = A4 + EQ(I,4)
      PC = PC + CP(I)
  225 CONTINUE
COMMENT:     GENERATE AREAS
      BASE = TT(M) - TT(1)
      TAE = (TAE/M) * BASE
      A1 = (A1/M) * BASE
      A2 = (A2/M) * BASE
      A3 = (A3/M) * BASE
      A4 = (A4/M) * BASE
      PC = (PC/M) * BASE
      TOTAL = A1 + A2 + A3 + A4 +PC
      XTAL = (A1 + A2 + A3 + A4) / TOTAL

      PRINT10,TAE,TOTAL
   10 FORMAT(1H ,//,1X,47HTOTAL AREA UNDER MEASURED INTENSITY ENVELOPE =
     1 ,F14.4,10X,15H(TOTAL PEAKS = ,F14.4,1H))
      PRINT11,A1,A2,A3,A4,A5
   11 FORMAT(1H0,28HAREAS OF DIFFRACTION PEAKS: ,5(F14.4,3X))
      PRINT12,PC
   12 FORMAT(1H0,34HAREA OF PARACRYSTALLINE SCATTER = ,F14.4)
      PRINT13,XTAL
   13 FORMAT(1H0,23HCRYSTALLINE FRACTION = ,F14.4)
      RETURN
      END
      SUBROUTINE MARQ(M,N,X,C,EPS,NSIG,ITMAX,FX,F,WK,IERR)             MARQ    2
C                                                                      MARQ   90
      DIMENSION WK(5),X(1),FX(1)                                       MARQ   91
      EXTERNAL F                                                       MARQ   92
      INDWK1 = 5*N+2*M+(N+1)*N/2+1                                     MARQ   96
      INDWK2 = INDWK1+1                                                MARQ   97
      INDWK3 = INDWK2+M*N                                              MARQ   98
      IOPT = 0                                                         MARQ   99
      DELTA = 0.0                                                      MARQ  100
      MAXFN = (ITMAX+1) * (N+1)                                        MARQ  101
      CALL QXZ031(F,M,N,NSIG,EPS,DELTA,MAXFN,IOPT,C,X,WK(INDWK1),FX,   QXZNA270
     *WK(INDWK2),M,WK(INDWK3),WK,INFER,IERR)                           MARQ  103
      ITMAX = WK(5)                                                    MARQ  104
      RETURN                                                           MARQ  105
      END                                                              MARQ  106
      SUBROUTINE  QXZ031(F,M,N,NSIG,EPS,DELTA,MAXFN,IOPT,PARM,X,SSQ,   QXZNA273
```

```
     *                    FX,XJAC,IXJAC,XJTJ,WK,INFER,IERR)                ZXSSQL 3
C                               THE SOLUTION X IS A STATIONARY POINT.      ZXSSQ150
C                                                                          ZXSSQ170
      DIMENSION           X(N),FX(M),PARM(5),XJAC(1),XJTJ(1),WK(6)         ZXSSQ171
C                                   XJAC USED INTERNALLY IN PACKED FORM    ZXSSQ172
      REAL                AL,CONS2,DELTA,DNORM,DSQ,EPS,                    ZXSSQ173
     *                    ERL2,ERL2X,FX,FO,FOSQ,FOSQS4,G,HALF,             ZXSSQ174
     *                    XJTJ,HH,ONE,ONEP10,ONEP5,ONESFO,AX,              ZXSSQ175
     *                    PREC,REL,RHH,SIG,SQDIF,SSQ,SSQOLD,SUM,TEN,       ZXSSQ176
     *                    TENTH,X,XDIF,XHOLD,XJAC,UP,WK,ZERO,              ZXSSQ177
     *                    XDABS,RELCON,PARM,P01,TWO,HUNTW,DELTA2           ZXSSQ178
      EXTERNAL F                                                           ZXSSQ179
      DATA                SIG/14.4/                                        ZXSSQ180
      DATA                AX/0.1/                                          ZXSSQ181
      DATA                P01,TENTH,HALF,ZERO,ONE,ONEP5,TWO,               ZXSSQ182
     *                    TEN,HUNTW,ONEP10/0.01,0.1,0.5,0.0,               ZXSSQ183
     *                    1.,1.5,2.,10.0,1.2E2,1.E10/                      ZXSSQ184
C                                   ERROR CHECKS                           ZXSSQ185
      IERR = 0                                                             ZXSSQ186
      IF (M.LE.0.OR.M.GT.IXJAC.OR.N.LE.0.OR.IOPT.LT.0.OR.IOPT.GT.2         ZXSSQ187
     C      .OR. PARM(1) .LE. 0.0) GO TO 305                               ZXSSQ188
      IMJC = IXJAC-M                                                       ZXSSQ189
      IF (IOPT.NE.2) GO TO 5                                               ZXSSQ190
      IF (PARM(2).LE.ONE.OR.PARM(1).LE.ZERO) GO TO 305                     ZXSSQ191
C                                   MACHINE DEPENDENT CONSTANTS            ZXSSQ192
    5 PREC = TEN**(-SIG-ONE)                                               ZXSSQ193
      REL = TEN**(-SIG*HALF)                                               ZXSSQ194
      RELCON = TEN**(-NSIG)                                                ZXSSQ195
C                                   WORK VECTOR IS CONCATENATION OF        ZXSSQ196
C                                   SCALED HESSIAN,GRADIENT,DELX,SCALE,    ZXSSQ197
C                                   XNEW,XBAD,F(X+DEL),F(X-DEL)            ZXSSQ198
      IGRAD1 = ((N+1)*N)/2                                                 ZXSSQ199
      IGRADL = IGRAD1+1                                                    ZXSSQ200
      IGRADU = IGRAD1+N                                                    ZXSSQ201
      IDELX1 = IGRADU                                                      ZXSSQ202
      IDELXL = IDELX1+1                                                    ZXSSQ203
      IDELXU = IDELX1+N                                                    ZXSSQ204
      ISCAL1 = IDELXU                                                      ZXSSQ205
      ISCALL = ISCAL1+1                                                    ZXSSQ206
      ISCALU = ISCAL1+N                                                    ZXSSQ207
      IXNEW1 = ISCALU                                                      ZXSSQ208
      IXNEWL = IXNEW1+1                                                    ZXSSQ209
      IXBAD1 = IXNEW1+N                                                    ZXSSQ210
      IFPL1 = IXBAD1+N                                                     ZXSSQ211
      IFPL = IFPL1+1                                                       ZXSSQ212
      IFPU = IFPL1+M                                                       ZXSSQ213
      IFML1 = IFPU                                                         ZXSSQ214
      IFML = IFML1+1                                                       ZXSSQ215
      IMJC = IXJAC - M                                                     ZXSSQ216
C                                   INITIALIZE VARIABLES                   ZXSSQ217
      AL = PARM(1)                                                         ZXSSQ218
      CONS2 = 0.0                                                          ZXSSQ219
      IF (IOPT.EQ.0) GO TO 20                                              ZXSSQ220
      IF (IOPT.EQ.1) GO TO 10                                              ZXSSQ221
```

```
      AL = PARM(1)
      FO = PARM(2)
      UP = PARM(3)
      CONS2 = PARM(4)*HALF
      GO TO 15
   10 AL = P01
      FO = TWO
      UP = HUNTW
      CONS2 = TENTH
   15 ONESFO = ONE/FO
      FOSQ = FO*FO
      FOSQS4 = FOSQ**4
   20 IEVAL = 0
      DELTA2 = DELTA*HALF
      ERL2 = ONEP10
      IBAD = -99
      ISW = 1
      ITER = -1
      INFER = 0
      IERR = 0
      DO 25 J=IDELXL,IDELXU
         WK(J) = ZERO
   25 CONTINUE
      GO TO 165
C                                 MAIN LOOP
   30 SSQOLD = SSQ
C                                 CALCULATE JACOBIAN
      IF (INFER.GT.0.OR.IJAC.GE.N.OR.IOPT.EQ.0.OR.ICOUNT.GT.0) GO TO 55
C                                 RANK ONE UPDATE TO JACOBIAN
      IJAC = IJAC+1
      DSQ = ZERO
      DO 35 J=IDELXL,IDELXU
         DSQ = DSQ + WK(J) * WK(J)
   35 CONTINUE
      IF (DSQ.LE.ZERO) GO TO 55
      DO 50 I=1,M
         G = FX(I) - WK(IFML1 + I)
         K = I
         DO 40 J=IDELXL,IDELXU
            G = G + XJAC(K) * WK(J)
            K = K+IXJAC
   40    CONTINUE
         G = G/DSQ
         K = I
         DO 45 J=IDELXL,IDELXU
            XJAC(K) = XJAC(K) - G * WK(J)
            K = K+IXJAC
   45    CONTINUE
   50 CONTINUE
      GO TO 80
C                                 JACOBIAN BY INCREMENTING X
   55 IJAC = 0
      K = -IMJC
      DO 75 J=1,N
```

```
            K = K+IMJC                                                          ZXSSQ276
            XDABS = ABS(X(J))                                                   ZXSSQ277
            HH = REL*(AMAX1(XDABS,AX))                                          ZXSSQ278
            XHOLD = X(J)                                                        ZXSSQ279
            X(J) = X(J)+HH                                                      ZXSSQ280
         CALL F(X,M,N,WK(IFPL))                                                 ZXSSQ281
         IEVAL = IEVAL +1                                                       ZXSSQ282
            X(J) = XHOLD                                                        ZXSSQ283
            IF (ISW.EQ.1) GO TO 65                                              ZXSSQ284
C                                  CENTRAL DIFFERENCES                          ZXSSQ285
            X(J) = XHOLD-HH                                                     ZXSSQ286
         CALL F(X,M,N,WK(IFML))                                                 ZXSSQ287
         IEVAL = IEVAL + 1                                                      ZXSSQ288
            X(J) = XHOLD                                                        ZXSSQ289
            RHH = HALF/HH                                                       ZXSSQ290
            DO 60 I=IFPL,IFPU                                                   ZXSSQ291
               K = K+1                                                          ZXSSQ292
               XJAC(K) = (WK(I) - WK(I + M)) * RHH                              ZXSSQ293
   60       CONTINUE                                                            ZXSSQ294
            GO TO 75                                                            ZXSSQ295
C                                  FORWARD DIFFERENCES                          ZXSSQ296
   65       RHH = ONE/HH                                                        ZXSSQ297
            DO 70 I=1,M                                                         ZXSSQ298
               K = K+1                                                          ZXSSQ299
               XJAC(K) = (WK(IFPL1 + I) - FX(I)) * RHH                          ZXSSQ300
   70       CONTINUE                                                            ZXSSQ301
   75 CONTINUE                                                                  ZXSSQ302
C                                  CALCULATE GRADIENT                           ZXSSQ303
   80 ERL2X = ERL2                                                              ZXSSQ304
      ERL2 = ZERO                                                               ZXSSQ305
      K = -IMJC                                                                 ZXSSQ306
      DO 90 J=IGRADL,IGRADU                                                     ZXSSQ307
         K = K+IMJC                                                             ZXSSQ308
         SUM = ZERO                                                             ZXSSQ309
         DO 85 I=1,M                                                            ZXSSQ310
            K = K+1                                                             ZXSSQ311
            SUM = SUM+XJAC(K)*FX(I)                                             ZXSSQ312
   85    CONTINUE                                                               ZXSSQ313
         WK(J) = SUM                                                            ZXSSQ314
         ERL2 = ERL2+SUM*SUM                                                    ZXSSQ315
   90 CONTINUE                                                                  ZXSSQ316
      ERL2 = SQRT(ERL2)                                                         ZXSSQ317
C                                  CONVERGENCE TEST FOR NORM OF GRADIENTZXSSQ318
      IF (IJAC.GT.0) GO TO 95                                                   ZXSSQ319
      IF (ERL2.LE.DELTA2) INFER = INFER+4                                       ZXSSQ320
      IF (ERL2.LE.CONS2) ISW = 2                                                ZXSSQ321
C                                  CALCULATE THE LOWER SUPER TRIANGE OF ZXSSQ322
C                                  JACOBIAN (TRANSPOSED) * JACOBIAN     ZXSSQ323
   95 L = 0                                                                     ZXSSQ324
      IS = -IXJAC                                                               ZXSSQ325
      DO 110 I=1,N                                                              ZXSSQ326
         IS = IS+IXJAC                                                          ZXSSQ327
         JS = -IXJAC                                                            ZXSSQ328
         DO 105 J=1,I                                                           ZXSSQ329
```

```
            JS = JS+IXJAC                                                  ZXSSQ330
            L = L+1                                                        ZXSSQ331
            SUM = ZERO                                                     ZXSSQ332
            DO 100 K=1,M                                                   ZXSSQ333
               LI = IS+K                                                   ZXSSQ334
               LJ = JS+K                                                   ZXSSQ335
               SUM = SUM+XJAC(LI)*XJAC(LJ)                                 ZXSSQ336
  100       CONTINUE                                                       ZXSSQ337
            XJTJ(L) = SUM                                                  ZXSSQ338
  105    CONTINUE                                                          ZXSSQ339
  110 CONTINUE                                                             ZXSSQ340
C                                  CONVERGENCE CHECKS                      ZXSSQ341
      IF (INFER.GT.0) GO TO 315                                            ZXSSQ342
      IF(IEVAL .GE. MAXFN) GO TO 295                                       ZXSSQ343
C                                  COMPUTE SCALING VECTOR                  ZXSSQ344
      IF (IOPT.EQ.0) GO TO 120                                             ZXSSQ345
      K = 0                                                                ZXSSQ346
      DO 115 J=1,N                                                         ZXSSQ347
         K = K+J                                                           ZXSSQ348
         WK(ISCAL1 + J) = XJTJ(K)                                          ZXSSQ349
  115 CONTINUE                                                             ZXSSQ350
      GO TO 135                                                            ZXSSQ351
C                                  COMPUTE SCALING VECTOR AND NORM         ZXSSQ352
  120 DNORM = ZERO                                                         ZXSSQ353
      K = 0                                                                ZXSSQ354
      DO 125 J=1,N                                                         ZXSSQ355
         K = K+J                                                           ZXSSQ356
         WK(ISCAL1 + J) = SQRT(XJTJ(K))                                    ZXSSQ357
         DNORM = DNORM+XJTJ(K)*XJTJ(K)                                     ZXSSQ358
  125 CONTINUE                                                             ZXSSQ359
      DNORM = ONE/SQRT(DNORM)                                              ZXSSQ360
C                                  NORMALIZE SCALING VECTOR                ZXSSQ361
      DO 130 J=ISCALL,ISCALU                                               ZXSSQ362
         WK(J) = WK(J) * DNORM * ERL2                                      ZXSSQ363
  130 CONTINUE                                                             ZXSSQ364
C                                  ADD L-M FACTOR TO DIAGONAL              ZXSSQ365
  135 ICOUNT = 0                                                           ZXSSQ366
  140 K = 0                                                                ZXSSQ367
      DO 150 I=1,N                                                         ZXSSQ368
         DO 145 J=1,I                                                      ZXSSQ369
            K = K+1                                                        ZXSSQ370
            WK(K) = XJTJ(K)                                                ZXSSQ371
  145    CONTINUE                                                          ZXSSQ372
  147    WK(K) = WK(K) + WK(ISCAL1 + I) * AL                               ZXSSQ373
  148    WK(IDELX1 + I) = WK(IGRAD1 + I)                                   ZXSSQ374
  150 CONTINUE                                                             ZXSSQ375
C                                  CHOLESKY DECOMPOSITION                  ZXSSQ376
  155 CALL QXZ032 (WK,1,N,WK(IDELXL),N,0,G,XHOLD,IERR)                     QXZNA276
      IF(IERR.EQ.0) GO TO 160                                              ZXSSQ378
      IERR = 0                                                             ZXSSQ379
      IF (IJAC.GT.0) GO TO 55                                              ZXSSQ380
      IF (IBAD.LE.0) GO TO 240                                             ZXSSQ381
      IF (IBAD.GE.2) GO TO 310                                             ZXSSQ382
      GO TO 190                                                            ZXSSQ383
```

```
  160 IF (IBAD.NE.-99) IBAD = 0
C                                  CALCULATE SUM OF SQUARES
  165 DO 170 J=1,N
      WK(IXNEW1 + J) = X(J) - WK(IDELX1 + J)
  170 CONTINUE
      CALL F(WK(IXNEWL),M,N,WK(IFPL))
      IEVAL = IEVAL + 1
      SSQ = ZERO
      DO 175 I=IFPL,IFPU
         SSQ = SSQ + WK(I) * WK(I)
  175 CONTINUE
  177 IF (ITER.GE.0) GO TO 185
C                                  SSQ FOR INITIAL ESTIMATES OF X
      ITER = 0
      SSQOLD = SSQ
      DO 180 I=1,M
         FX(I) = WK(IFPL1 + I)
  180 CONTINUE
      GO TO 55
  185 IF (IOPT.EQ.0) GO TO 215
C                                  CHECK DESCENT PROPERTY
      IF (SSQ.LE.SSQOLD) GO TO 205
C                                  INCREASE PARAMETER AND TRY AGAIN
  190 ICOUNT = ICOUNT+1
      AL = AL*FOSQ
      IF (IJAC.EQ.0) GO TO 195
      IF (ICOUNT.GE.4.OR.AL.GT.UP) GO TO 200
  195 IF (AL.LE.UP) GO TO 140
      IF (IBAD.EQ.1) GO TO 310
      GO TO 300
  200 AL = AL/FOSQS4
      GO TO 55
C                                  ADJUST MARQUARDT PARAMETER
  205 IF (ICOUNT.EQ.0) AL = AL/FO
      IF (ERL2X.LE.ZERO) GO TO 210
      G = ERL2/ERL2X
      IF (ERL2.LT.ERL2X) AL = AL*AMAX1(ONESFO,G)
      IF (ERL2.GT.ERL2X) AL = AL*AMIN1(FO,G)
  210 AL = AMAX1(AL,PREC)
C                                  ONE ITERATION CYCLE COMPLETED
  215 ITER = ITER+1
      DO 220 J=1,N
         X(J) = WK(IXNEW1 + J)
  220 CONTINUE
      DO 225 I=1,M
         WK(IFML1 + I) = FX(I)
         FX(I) = WK(IFPL1 + I)
  225 CONTINUE
C                                  RELATIVE CONVERGENCE TEST FOR X
      IF (ICOUNT.GT.0.OR.IJAC.GT.0) GO TO 30
      DO 230 J=1,N
         XDIF = ABS(WK(IDELX1 + J)) / AMAX1(ABS(X(J)),AX)
         IF (XDIF.GT.RELCON) GO TO 235
  230 CONTINUE
```

```
      INFER = INFER + 2                                                 ZXSSQ438
C                                  RELATIVE CONVERGENCE TEST FOR SSQ    ZXSSQ439
  235 SQDIF = ABS(SSQ-SSQOLD)/AMAX1(SSQOLD,AX)                          ZXSSQ440
      IF (SQDIF .LE. EPS) INFER = INFER+1                               ZXSSQ441
      IF(IBAD .EQ. -99) GO TO 30                                        ZXSSQ442
      IF(INFER .NE. 0 .AND. SSQ .GT. (10.0 * EPS)) GO TO 310            ZXSSQ443
      GO TO 30                                                          ZXSSQ444
C                                  SINGULAR DECOMPOSITION               ZXSSQ445
  240 IF (IBAD) 255,245,265                                             ZXSSQ446
C                                  CHECK TO SEE IF CURRENT              ZXSSQ447
C                                  ITERATE HAS CYCLED BACK TO           ZXSSQ448
C                                  THE LAST SINGULAR POINT              ZXSSQ449
  245 DO 250 J=1,N                                                      ZXSSQ450
         XHOLD = WK(IXBAD1 + J)                                         ZXSSQ451
         IF (ABS(X(J)-XHOLD).GT.RELCON*AMAX1(AX,ABS(XHOLD))) GO TO 255  ZXSSQ452
  250 CONTINUE                                                          ZXSSQ453
      GO TO 310                                                         ZXSSQ454
C                                  UPDATE THE BAD X VALUES              ZXSSQ455
  255 DO 260 J=1,N                                                      ZXSSQ456
         WK(IXBAD1 + J) = X(J)                                          ZXSSQ457
  260 CONTINUE                                                          ZXSSQ458
      IBAD = 1                                                          ZXSSQ459
C                                  INCREASE DIAGONAL OF HESSIAN         ZXSSQ460
  265 IF (IOPT.NE.0) GO TO 280                                          ZXSSQ461
      K = 0                                                             ZXSSQ462
      DO 275 I=1,N                                                      ZXSSQ463
         DO 270 J=1,I                                                   ZXSSQ464
            K = K+1                                                     ZXSSQ465
            WK(K) = XJTJ(K)                                             ZXSSQ466
  270    CONTINUE                                                       ZXSSQ467
         WK(K) = ONEP5 * (XJTJ(K) + AL * ERL2 * WK(ISCAL1 + I)) + REL   ZXSSQ468
  275 CONTINUE                                                          ZXSSQ469
      IBAD = 2                                                          ZXSSQ470
      GO TO 155                                                         ZXSSQ471
C                                  REPLACE ZEROES ON HESSIAN DIAGONAL   ZXSSQ472
  280 IZERO = 0                                                         ZXSSQ473
      DO 285 J=ISCALL,ISCALU                                            ZXSSQ474
         IF(WK(J) . GT . ZERO) GO TO 285                                ZXSSQ475
         IZERO = IZERO+1                                                ZXSSQ476
         WK(J) = ONE                                                    ZXSSQ477
  285 CONTINUE                                                          ZXSSQ478
      IF (IZERO.LT.N) GO TO 140                                         ZXSSQ479
C                                  TERMINAL ERROR                       ZXSSQ480
  290 IERR = IERR + 1                                                   ZXSSQ481
  295 IERR = IERR + 1                                                   ZXSSQ482
  300 IERR = IERR + 1                                                   ZXSSQ483
  305 IERR = IERR + 1                                                   ZXSSQ484
  310 CONTINUE                                                          ZXSSQ485
      IERR = IERR + 1                                                   ZXSSQ486
      IF (IERR . EQ . 2) GO TO 9005                                     ZXSSQ487
C                                  OUTPUT ERL2,IEVAL,NSIG,AL, AND ITER  ZXSSQ488
  315 G = SIG                                                           ZXSSQ489
      DO 320 J=1,N                                                      ZXSSQ490
         XHOLD = ABS(WK(IDELX1 + J))                                    ZXSSQ491
```

```
         IF (XHOLD.LE.ZERO) GO TO 320                                  ZXSSQ492
         G = AMIN1(G,-ALOG10(XHOLD)+ALOG10(AMAX1(AX,ABS(X(J)))))       ZXSSQ493
  320 CONTINUE                                                         ZXSSQ494
      IF(N.GT.2) GO TO 330                                             ZXSSQ495
      DO 325 J = 1,N                                                   ZXSSQ496
  325 WK(J+5) = WK(J+IGRAD1)                                           ZXSSQ497
  330 WK(1) = ERL2+ERL2                                                ZXSSQ498
      WK(2) = IEVAL                                                    ZXSSQ499
      SSQ = SSQOLD                                                     ZXSSQ500
      WK(3) = G                                                        ZXSSQ501
      WK(4) = AL                                                       ZXSSQ502
      WK(5) = ITER                                                     ZXSSQ503
 9005 RETURN                                                           ZXSSQ504
      END                                                              ZXSSQ505
      SUBROUTINE QXZ032 (A,M,N,B,IB,IDGT,D1,D2,IER)                    QXZNA279
C                                                                      LEQT1P 3
C   LANGUAGE            - FORTRAN                                      LEQT1P39
C                                                                      LEQT1P42
      DIMENSION          A(1),B(IB,1)                                  LEQT1P43
C                                  INITIALIZE IER                      LEQT1P44
      IER = 0                                                          LEQT1P45
C                                  DECOMPOSE A                         LEQT1P46
      CALL QXZ033 (A,A,N,D1,D2,IER)                                    QXZNA282
      IF (IER.NE.0) GO TO 9005                                         LEQT1P48
C                                  PERFORM ELIMINATION                 LEQT1P49
      DO 5 I = 1,M                                                     LEQT1P50
         CALL QXZ034 (A,B(1,I),N,B(1,I))                               QXZNA283
    5 CONTINUE                                                         LEQT1P52
 9005 RETURN                                                           LEQT1P53
      END                                                              LEQT1P54
      SUBROUTINE QXZ033 (A,UL,N,D1,D2,IER)                             QXZNA286
C                                                                      LUDECP 3
C   PRECISION           - SINGLE                                       LUDECP25
      DIMENSION          A(1),UL(1)                                    LUDECP31
      DATA               ZERO,ONE,FOUR,SIXTN,SIXTH/0.0,1.,4.,16.,.0625/ LUDECP32
      D1=ONE                                                           LUDECP33
      D2=ZERO                                                          LUDECP34
      RN = ONE/(N*SIXTN)                                               LUDECP35
      IP = 1                                                           LUDECP36
      IER=0                                                            LUDECP37
      DO 45 I = 1,N                                                    LUDECP38
         IQ = IP                                                       LUDECP39
         IR = 1                                                        LUDECP40
         DO 40 J = 1,I                                                 LUDECP41
            X = A(IP)                                                  LUDECP42
            IF (J .EQ. 1) GO TO 10                                     LUDECP43
            DO 5  K=IQ,IP1                                             LUDECP44
               X = X-UL(K)*UL(IR)                                      LUDECP45
               IR = IR+1                                               LUDECP46
    5       CONTINUE                                                   LUDECP47
   10       IF (I.NE.J) GO TO 30                                       LUDECP48
            D1 = D1*X                                                  LUDECP49
            IF (A(IP)+X*RN .LE. A(IP)) GO TO 50                        LUDECP50
   15       IF (ABS(D1) .LE. ONE) GO TO 20                             LUDECP51
```

```
          D1 = D1 * SIXTH                                               LUDECP52
          D2 = D2 + FOUR                                                LUDECP53
          GO TO 15                                                      LUDECP54
   20     IF (ABS(D1) .GE. SIXTH) GO TO 25                              LUDECP55
          D1 = D1 * SIXTN                                               LUDECP56
          D2 = D2 - FOUR                                                LUDECP57
          GO TO 20                                                      LUDECP58
   25     UL(IP) = ONE/SQRT(X)                                          LUDECP59
          GO TO 35                                                      LUDECP60
   30     UL(IP) = X * UL(IR)                                           LUDECP61
   35     IP1 = IP                                                      LUDECP62
          IP = IP+1                                                     LUDECP63
          IR = IR+1                                                     LUDECP64
   40   CONTINUE                                                        LUDECP65
   45 CONTINUE                                                          LUDECP66
      GO TO 9005                                                        LUDECP67
   50 IER = 129                                                         LUDECP68
 9000 CONTINUE                                                          LUDECP69
 9005 RETURN                                                            LUDECP70
      END                                                               LUDECP71
      SUBROUTINE QXZ034 (A,B,N,X)                                       QXZNA290
C                                                                       LUELMP 3
C                                                                       LUELMP25
      DIMENSION          A(1),B(1),X(1)                                 LUELMP26
      DATA               ZERO/0./                                       LUELMP27
C                                  SOLUTION OF LY = B                   LUELMP28
      IP=1                                                              LUELMP29
      IW = 0                                                            LUELMP30
      DO 15 I=1,N                                                       LUELMP31
         T=B(I)                                                         LUELMP32
         IM1 = I-1                                                      LUELMP33
         IF (IW .EQ. 0) GO TO 9                                         LUELMP34
         IP=IP+IW-1                                                     LUELMP35
         DO 5 K=IW,IM1                                                  LUELMP36
            T = T-A(IP)*X(K)                                            LUELMP37
            IP=IP+1                                                     LUELMP38
    5    CONTINUE                                                       LUELMP39
         GO TO 10                                                       LUELMP40
    9    IF (T .NE. ZERO) IW = I                                        LUELMP41
         IP = IP+IM1                                                    LUELMP42
   10    X(I)=T*A(IP)                                                   LUELMP43
         IP=IP+1                                                        LUELMP44
   15 CONTINUE                                                          LUELMP45
C                                  SOLUTION OF UX = Y                   LUELMP46
      N1 = N+1                                                          LUELMP47
      DO 30 I = 1,N                                                     LUELMP48
         II = N1-I                                                      LUELMP49
         IP=IP-1                                                        LUELMP50
         IS=IP                                                          LUELMP51
         IQ=II+1                                                        LUELMP52
         T=X(II)                                                        LUELMP53
         IF (N.LT.IQ) GO TO 25                                          LUELMP54
         KK = N                                                         LUELMP55
         DO 20 K=IQ,N                                                   LUELMP56
```

```
          T = T-A(IS)*X(KK)                                     LUELMP57
          KK = KK-1                                             LUELMP58
          IS = IS-KK                                            LUELMP59
   20   CONTINUE                                                LUELMP60
   25   X(II)=T*A(IS)                                           LUELMP61
   30 CONTINUE                                                  LUELMP62
      RETURN                                                    LUELMP63
      END                                                       LUELMP64
```

1 Report No	2 Government Accession No	3 Recipient's Catalog No
NASA TM-86412		

4 Title and Subtitle	5 Report Date
Crystallinity Determination by Curvefit Procedure for a Semi-crystalline Polymer	May 1985
	6 Performing Organization Code 505-33-33-09
7 Author(s) N. T. Wakelyn	8 Performing Organization Report No
	10 Work Unit No
9 Performing Organization Name and Address NASA Langley Research Center Hampton, VA 23665	11 Contract or Grant No
	13 Type of Report and Period Covered Technical Memorandum
12 Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546	14 Sponsoring Agency Code

15 Supplementary Notes

16 Abstract

Wide angle x-ray scattering (WAXS) data from poly(etheretherketone) (PEEK) has been resolved into a crystalline contribution represented as 4 reflections and an amorphous contribution represented as a broad, smoothly varying curve, both contributions occurring in the 2θ range: 15-31 degrees. In this resolution the crystalline scatter is described as a linear combination of Cauchy and Gaussian functions while that of the amorphous halo is expressed as a cubic polynomial. Statistical analysis of the measured scattered intensity from an amorphous specimen with that calculated from the cubic polynomial, as a function of the combination parameter (fraction of Cauchy and Gaussian functions), suggests that the crystalline fraction of the polymer specimen studied is about 0.39. A listing of the FORTRAN IV program used in the resolution is provided in the Appendix.

17 Key Words (Suggested by Author(s))	18 Distribution Statement
Crystallinity X-Ray Diffraction Thermoplastic Poly(etheretherketone)	Unclassified - Unlimited Subject Category 27

19 Security Classif (of this report)	20 Security Classif (of this page)	21 No of Pages	22 Price
Unclassified	Unclassified	34	A03

End of Document